

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-073405
(43)Date of publication of application : 18.03.1997

(51)Int.Cl.

G06F 11/32
G06F 3/14

(21)Application number : 07-229889
(22)Date of filing : 07.09.1995

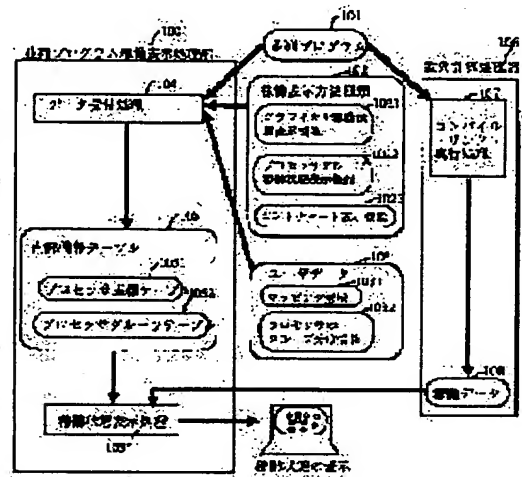
(71)Applicant : HITACHI LTD
(72)Inventor : SAJI MIYUKI

(54) DISPLAY METHOD FOR PARALLEL PROGRAM OPERATION STATE

(57)Abstract:

PROBLEM TO BE SOLVED: To make it easy to grasp the operation state of programs and also grasp the operation of all the programs by arranging a processor in a user's image shape and displaying the performance of the processor.

SOLUTION: A program operation display processing part 100 when receiving input data 101-103 through data reception processing 104 generates an internal information table 105 corresponding to the user data 103 first. Then the processing is moved to the parallel computer processing part 105 of a parallel computer and a parallel program 101 is executed through compilation and linking execution processing 107. While the program is executed, operation data 108 as performance data on respective processors and memories are outputted. The program and operation display processing part 100 on receiving the operation data 108 displays operation states next through operation display processing 109 according to various operation display kinds 102.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-73405

(43)公開日 平成9年(1997)3月18日

(51)Int.Cl.⁵

G 0 6 F 11/32
3/14

識別記号

3 2 0

庁内整理番号

7313-5B

F I

G 0 6 F 11/32
3/14

A

3 2 0 A

技術表示箇所

審査請求 未請求 請求項の数 4 O L (全 7 頁)

(21)出願番号 特願平7-229889

(22)出願日 平成7年(1995)9月7日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 佐治 みゆき

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

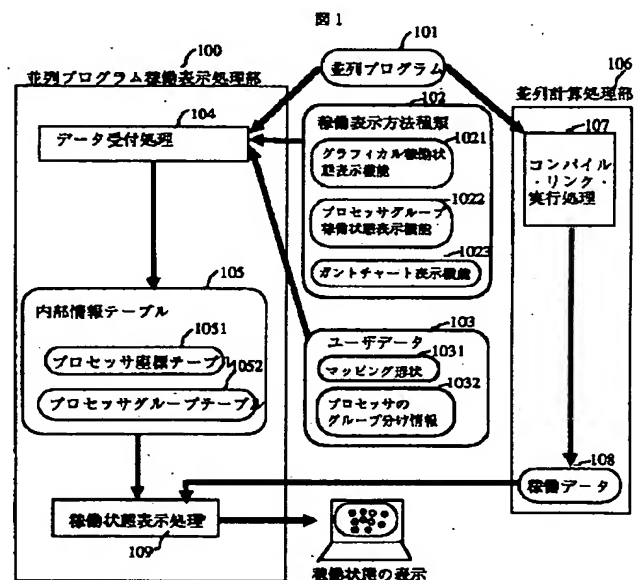
(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 並列プログラム稼働状態表示方法

(57)【要約】

【構成】本並列プログラム稼働表示処理部100は、並列プログラム101と稼働表示方法種類102とユーザデータ103を受け付け、内部情報テーブル105を生成するデータ受付処理104と、並列プログラム101を実行して得られた稼働データ108と、さきの内部情報テーブル105より目的の稼働表示方法で並列プログラムの実行状態を表示する稼働状態表示処理109よりなる。

【効果】ユーザの指定した形状にプロセッサを配置して並列計算機の稼働状況を表示できるため、プログラムの構造に沿った表示が可能である。また、プロセッサをグループに分けてグループ単位に稼働状況を表示できるため、大規模プログラムでもその稼働状態を一度にみることができ、並列プログラムのデバッグ、最適化の工数を削減できる。



【特許請求の範囲】

【請求項1】並列計算機の各プロセッサから得られた情報を、ユーザが指定した形状にプロセッサを配置し、その稼働状態を表示することを特徴とする並列プログラム稼働状態表示方法。

【請求項2】並列計算機の各プロセッサから得られた情報をユーザの指定した複数のグループに分け、グループ単位に前記情報を表示することを特徴とする並列プログラム稼働状態表示方法。

【請求項3】並列計算機の各プロセッサから得られた情報をユーザの指定した複数のグループに分け、ユーザ指定した形状に前記プロセッサのグループを配置して、グループ単位に前記情報を表示する請求項第1または2に記載の並列プログラム稼働状態表示方法。

【請求項4】並列プログラムの実行状態表示方法において、全プロセッサをユーザの指定した複数のグループに分け、1グループが1プロセッサのように稼働状況を表示することを特徴とする並列プログラム稼働状態表示方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】本発明は任意の並列プログラムを対象とし、この並列プログラムの稼働状況の表示方法に関する。

【0002】

【従来の技術】従来技術として以下に二つ示す。一つは計算機性能そのものの表示技術である。

【0003】並列計算機の性能を引き出すためには、その上で動くプログラムが複数のプロセッサを偏り無く利用し、かつ各CPU、I/O、ネットワークに極端に負荷の大きい場所（ボトルネックという）を発生させないことが重要である。通常、このような理想的な並列プログラムを利用者が初めから記述することは容易ではない。したがって、まず論理的に正しく稼働するプログラムを作成し、その並列機上での稼働状況を観察してチューニングを施し、最終的に高速に稼働するプログラムに仕上げるという手順がとられる。そのための手間を低減するためには、並列プログラムの稼働状況を利用者がすばやく把握できるツールが不可欠である。

【0004】これに対して、従来、計算機の性能データ（CPU稼働率、I/O利用状況、メモリ利用状況、ネットワーク利用状況など）を表示する方法は、インテル（Intel）社のパレード（Parade）で行われているように、並列計算機に対応する図をコンピュータスクリーン上に描画し、性能データの高低によって対応する部分の色を変更させて並列計算機の稼働状況を表示していた。これは並列計算機に対応して各プロセッサ一つの性能を表示するものである。

【0005】もう一つの従来技術は、さらに、並列プログラムのもっと詳細な実行状態を把握するものである。

メッセージパッシング型の並列計算機では、各プロセッサ毎にローカルメモリを持ち、必要な情報はプロセッサからプロセッサへ通信路を介して送受信される。

【0006】効率よい並列プログラムを作成するには、プログラム中のデータをどのように各メモリに配分するかが大きな要因となる。また、デバッグにおいてはこれら配置したデータが正しく動作しているか確認することが重要である。一般的なプログラム記述言語C、フォートラン（FORTRAN）などを用いる場合はデータ割り付けおよび通信に関する指示はプログラム上でユーザが記述する。具体的には、あるプロセッサからセンド（send）命令を用いて情報を送信し、指定されたプロセッサではレシーブ（receive）命令を用いてこれを受け取る、またあるプロセッサからブロードキャスト（broadcast）命令を用いてすべてのプロセッサへ情報を送信し、全プロセッサではこれをreceive命令を用いて受け取る、などが一般的である。このプロセッサ間通信はタイミングを制御してプログラミングしないと、無駄な待ち時間ができたり、デッドロックを起こしたりする。そして、並列プログラムに於てはこのプロセッサ間通信のタイミングを最適化することがプログラムの実行時間の短縮に、最終的にはプログラムの性能向上につながる。

【0007】このプロセッサ間通信に関する最適化を支援するための従来技術は、一つに、XPVM（Oak Ridge 国立研究所）など、ユーザプログラムを実行し、プロセッサ間のデータの送受信のタイミングを、縦軸をプロセッサ、横軸を時間で表すガントチャートなどを用いてグラフィカルに表示するものがある。これを用いれば送受信の時間や、待ち時間、タイミングなどが一目でわかるので、プログラムの修正すべき点が把握しやすく、プログラムの最適化をはかる際に有用である。

【0008】

【発明が解決しようとする課題】上記従来技術を用いると、並列計算機の性能や並列計算機の作成プログラムにおける各実行文のタイミングやデータ通信のタイミングの把握は可能である。しかし、以下の課題が残る。

【0009】（1）プロセッサの稼働状況がわかっても、それがプログラムの構造上どの役割の部分か把握するのが困難である。

【0010】（2）プロセッサの数が多い場合、全てのプロセッサの稼働状態を出力するとその並列計算機の動作の把握が非常に困難である。

【0011】（3）プロセッサの数が多い場合、全てのプロセッサのガントチャートを出力するとプログラムの動作の把握が非常に困難である。ガントチャートは時系列なデータが一目で把握できるため並列プログラムのデバッグ・チューニングには欠かせないツールであるが、並列機は少ない数十台のプロセッサでさえ、全プロセッサのガントチャートを出力すると情報量は多くなりプログラム動作の把握はむずかしい。

【0012】本発明の目的はこれらの課題を解決し、効率よく並列計算機性能および並列プログラムの稼働状態を表示する方法を提供することにある。

【0013】

【課題を解決するための手段】上記各課題を解決するための手段は以下の通りである。

【0014】(1)メッセージパッシング型の並列計算機を用いる際には、大きなデータを複数のプロセッサに割り付けプログラミングするケースが多く、このデータのメモリ割り付けがプログラミング上重要である。そしてこのデータはユーザが形状をイメージしていることが多い。そこで、プロセッサをこのユーザイメージ形状に配置してプロセッサの稼働状態を表示することにより、並列プログラム構造上に沿った稼働状態の表示が達成される。

【0015】(2)複数のプロセッサを1プロセッサとみて並列プログラムの性能を表示する、またプログラムの実行状態を示すガントチャートの表示においても複数のプロセッサの情報を1プロセッサの情報とみなして表示項目を削減することにより、少ない表示項目で大規模なプログラムの稼働状態の表示が達成される。

【0016】

【作用】上記手段に対する作用は以下の通りである。

【0017】(1)プロセッサをユーザイメージ形状に配置してプロセッサの性能を表示することにより、プログラムの稼働状態の把握が簡単になる。

【0018】(2)全てのプロセッサに関する情報を、少ない情報量で表示するため、プログラム全体の動作が簡単に把握できる。

【0019】

【実施例】図4は本発明の前提にするハードウェア構成である。ユーザにより稼働表示の対象である並列プログラム101はワークステーション上の並列プログラム稼働表示処理部100および並列計算機110に入力され処理される。並列計算機110には複数のcpu401があり、それぞれメモリ402およびデータ採集部403を持つ。この並列計算機110で並列プログラムを実行して得られた稼働データ108は通信路404をへて、並列プログラム稼働表示処理部100へ送られ、処理される。本発明はこの並列プログラム稼働表示処理部100の処理に関する。以下、この並列プログラム稼働表示処理部100および並列処理部106について詳細に説明する。

【0020】図1は本発明の実施例を示す並列プログラム稼働表示方法の処理手順である。並列プログラム稼働表示処理部100の入力は稼働表示の対象である並列プログラム101と、どのような表示を指定するかを示す稼働表示方法種類102、および各稼働表示方法において必要となるユーザデータ103である。稼働表示方法種類102は3種類あり、並列計算機の各プロセッサの

稼働状態をユーザの指定する形状にマッピングして表示するグラフィカルプロセッサ稼働状態表示機能1021と、複数のプロセッサを1プロセッサのようにして示すプロセッサグループ稼働状態表示機能1022およびガントチャートをプロセッサグループで表示するガントチャート機能1023である。それぞれに必要なユーザデータ103はグラフィカルプロセッサ稼働状態表示機能1021を指定した場合はプロセッサの表示形状を示すマッピング形状1031、プロセッサグループ稼働状態表示機能1022およびガントチャート機能1023を指定した場合はプロセッサのグループ分け情報1032である。

【0021】プログラム稼働表示処理部100では、データ受付処理104によって入力データ101、102、103を受けるとはじめに、ユーザデータ103に対応する内部情報テーブル105を生成する。内部情報テーブル105は具体的には、マッピング形状1031に対応したプロセッサ座標テーブル1051、プロセッサのグループ分け情報1032に対応したプロセッサグループテーブル1052を生成する。次に処理を並列計算機の並列計算機処理部106に移し、コンパイル・リンク・実行処理107によって並列プログラム101を実行する。プログラムを実行しながら、各プロセッサやメモリの性能データである稼働データ108を出力する。この稼働データ108を受けて、プログラム稼働表示処理部100では、次に、稼働表示処理109によりユーザの指定した各種稼働表示種類102に応じて稼働状況を表示する。具体的には、実行時に並列計算機処理部106から送られる稼働データ108を内部情報テーブル105にマッピングして、目的の表示を実施する。以上が、並列プログラム稼働表示方法100の処理手順の概略である。

【0022】以下、例を用いて本発明の詳細を示す。

【0023】入力情報および、最初の処理であるデータ受付処理104の詳細を図2を用いて示す。図2は入力情報に対する、データ受付画面201である。データ受付処理104はこの受付処理画面201によって対話型に入力情報101、102、103を受け付ける。

【0024】はじめに、稼働状態表示対象である並列プログラム101をファイル名称によって読み込む(202)。これはユーザが複数のプロセッサを対象にFORTRANやCなどのプログラミング言語で記述したものである。

【0025】次に、稼働表示方法種類102を選択形式で読み込む(203)。選択肢は三つでグラフィカルプロセッサ稼働状態表示機能1021とプロセッサグループ稼働状態表示機能1022とガントチャート機能1023で、ユーザは対応するトグルボタンを押下することによって選択する。グラフィカルプロセッサ稼働状態表示機能1021とプロセッサグループ稼働状態表示機能

1022は両方を押下することも可能である。本例ではこの二つを選択している。

【0026】次に選択された機能に従い、必要なデータを要求する。まず、グラフィカルプロセッサ稼働状態表示機能1021が指定された場合は、グラフィカルプロセッサ表示用データ受付204の画面を表示し、データを受け付ける。具体的にはまず、指定する形状が個々のプロセッサに対するものかプロセッサグループに対するものかトグルボタンによって選択する(205)。本例では上記でプロセッサグループ表示も選択しているため、プロセッサグループを選択する。

【0027】次に、マッピングすべき形状を入力する。画面右中にある四つの部品206から形状を選択しこれらを組み合わせて目的の形状を作成する(207)。部品中のn, m, p等はプロセッサ番号をユーザが入力する箇所である。部品中に一つの番号を入力する場合は1プロセッサまたはプロセッサグループを示し、二つの場合は連続するプロセッサ番号のはじめと終わりを意味する。例えばユーザ記入例207の一番上の長方形はプロセッサグループ番号1〜5までを示している。一つの四角形に四つの番号を入力する場合も連続するプロセッサ番号であり、途中の番号を二つ記入することで縦横どちらの方向にプロセッサを連続して配置するかを指示する。本例では6〜25までのプロセッサグループを横方向に五つずつ連続して配置したものを縦方向に並べるとを指示している。

【0028】また、プロセッサグループ稼働状態表示機能1022およびガントチャート機能1023が指示された場合はプロセッサグループ表示用データ受付208の画面を表示し、データを受け付ける。具体的にはグループ番号を記入し209それに付属するプロセッサ番号を記入する(210)。連続したプロセッサ番号は記号〜を用いて記述することができる。例えば本例ではグループ1にはプロセッサ番号1〜10が含まれることを指示している。

【0029】以上の方法を用いてユーザは入力データの指示を行い、終了したら終了ボタン212を押下する。データ受付処理102は終了ボタン212の押下によって入力データ101, 102, 103の受け付けを終了し、内部情報テーブル105として、プロセッサ座標テーブル1051およびプロセッサグループテーブル1052を生成する。

【0030】プロセッサ座標テーブル1051の例を図3に示す。プロセッサ座標テーブル105は全てのプロセッサまたはプロセッサグループ311に対してその座標値312を持つ。座標値はデータ受付処理104でユーザに指定された配置より求める。また、プロセッサグループテーブル1051では各グループ番号に付属する全てのプロセッサ番号を持つ。

【0031】次に処理を並列計算機110における、並

列計算機処理部106に移し、コンパイル・リンク・実行処理107によって並列プログラム101を実行する。プログラムを実行しながら、各プロセッサやメモリの性能データである稼働データ108を出力する。

【0032】図4を用いて説明する。並列計算機には複数のcpu401があり、それぞれメモリ402を持つ。また、それぞれデータ収集部403を持ち、性能のデータを収集する。収集するのは、例えば、プロセッサ、メモリ、通信系、受信系などの使用状態である。各性能データは各プロセッサ上で稼働するオペレーティングシステムにシステムコールを発行することにより得ることができる。それぞれ収集した性能データである稼働データ108はネットワーク404経由で、プログラム稼働表示処理部100のあるワークステーションなどの計算機に送られる。

【0033】並列計算機107から稼働データ108がプログラム稼働表示処理部100に送られると、稼働表示処理109によりユーザの指定した各種稼働表示種類102に応じて稼働状況を表示する。指定された稼働表示種類102がグラフィカルプロセッサ稼働状態表示機能1021の場合は、プロセッサ座標テーブル105にあわせてプロセッサを配置し、稼働データ108を表示する。また、プロセッサグループ稼働状態表示機能1022が指定されている場合は、稼働データ108をプロセッサグループテーブル106にあわせて統合し、これを表示する。

【0034】図2に示したユーザの入力に対する出力例を図5に示す。図5の画面表示例で上段(501)は、図2のユーザ指定のプロセッサグループおよびプロセッサグループ配置上で入力した並列プログラム101の稼働状態である。マウス507でプロセッサグループをクリックするとグループ名称とその所属プロセッサを表示する(502)。下段(503)には以下の選択項目を表示する。表示データとしてcpu稼働率、メモリ、通信のいずれかをユーザは選ぶ(504)。また、各プロセッサグループの番号を入力するとそのプロセッサグループの詳細を表示することも可能である(505)。上段の画面表示の視点を変更することもできる(506)。

【0035】本実施例によれば、以下の効果がある。

【0036】本プログラムは入力した形状と同一の物理形状を持つモデルの数値シミュレーションであるが、プロセスグループ1〜5の負荷が重く、21〜25のプロセスグループの稼働の負荷が軽い。よって、メモリ割り付けにおいてプロセスグループ1〜5に対応する物理領域を少なく、プロセスグループ21〜25に対応する物理領域を多くすればよいことが一目でわかる。

【0037】

【発明の効果】本発明を用いれば、ユーザの指定した形状にプロセッサを配置して並列計算機の稼働状況を表示

できる。よって、プログラムの構造に沿った表示が可能である。また、プロセッサをグループに分けてグループ単位に稼働状況を表示できるため、大量のプロセッサを駆動するプログラムでも、プログラム全体の稼働状態を一度にみることができる。よって本発明によれば、並列プログラムのデバッグ、最適化の工数を削減できる。

【図面の簡単な説明】

【図1】 本発明の実施例を示す並列プログラム稼働表示処理のフローチャート。

【図2】 データ受付処理画面の例の説明図。

【図3】 プロセッサ座標テーブルの例の説明図。

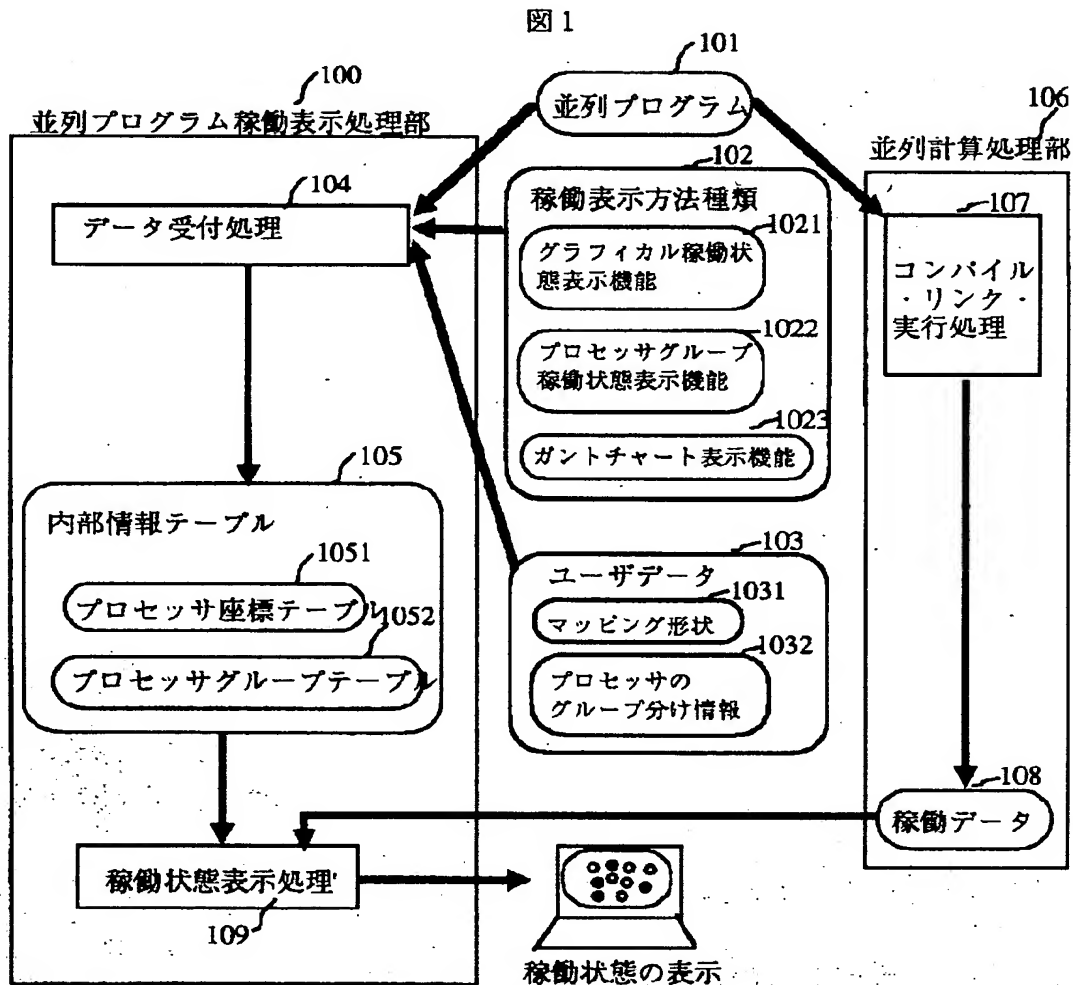
【図4】 ハードウェア構成の説明図。

【図5】 並列プログラム稼働状態表示例の説明図。

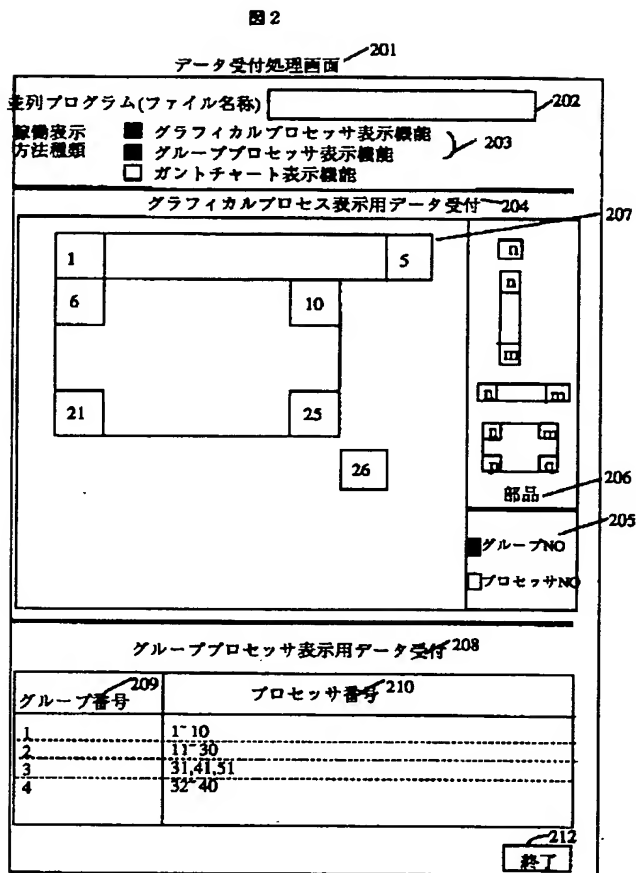
【符号の説明】

100…並列プログラム稼働表示処理部、101…並列プログラム、102…稼働表示方法種類、1021…グラフィカル稼働状態表示機能、1022…プロセッサグループ稼働状態表示機能、1023…ガントチャート表示機能、103…ユーザデータ、1031…マッピング形状、1032…プロセッサのグループ分け情報、104…データ受付処理、105…内部情報テーブル、1051…プロセッサ座標テーブル、1052…プロセッサグループテーブル、106…並列計算処理部、107…コンパイル・リンク・実行処理、108…稼働データ、109…稼働状態表示処理。

【図1】



【図2】



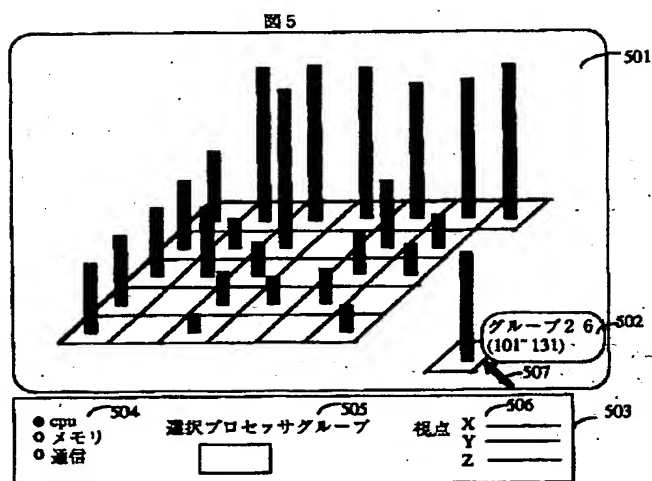
【図3】

図3

プロセッサ座標テーブル 1051

プロセッサ番号 311	x座標 312	y座標
1	0.1	0.1
2	1.1	0.1
3	2.1	0.1
4	3.1	0.1
5	4.1	0.1
6	0.1	1.1
.	.	.
.	.	.

【図5】



【図4】

図4

